Documentation for a Flask Application

RESTful API for managing a To-Do list.

# Introduction

This document provides the details about the To-Do List Flask API project, which is designed to manage a list of tasks and to-dos. The application will allow users to create, read, update and delete to-do items.

# Technology Stack

* Programming Language: Python
* Framework: Flask
  + The Flask Framework is used to create the API endpoints and manage the server-side logic.
* Database: Python list
  + The current “database” uses a Python list to store the to-do items. However, it lacks data persistence.

# Development Environment

* IDE: Visual Studio Code
* Version Control: GitHub
* Virtual Environment: Python’s built-in “venv” module.

# API Endpoints

The API provides the following endpoints:

1. GET ‘/todos’: Fetches all to-do items.
2. POST ‘/todos’: Creates a new to-do item.
3. GET ‘/todos/<id>’: Retrieves a specific to-do item by its ID.
4. PUT/PATCH ‘/todos/<id>’: Updates a specific to-do item. PUT replaces the entire item, PATCH updates specific fields.
5. DELETE ‘/todos/<id>’: Deletes a specific to-do item by its ID.

Each to-do item contains the following fields:

* Id
* Title
* Description
* Completed
* CreatedAt
* UpdatedAt

# Additional Supporting Tools

* Postman: Used to test API endpoint manually.
* ChatGPT: Used to set the direction to develop the Flask Application and to guide on the usage of the API.
* PythonAnywhere: Used to host the application.